
Testi del Syllabus

Resp. Did. **AMORETTI Michele**

Matricola: **207094**

Anno offerta: **2015/2016**

Insegnamento: **05924 - SISTEMI DI ELABORAZIONE**

Corso di studio: **5015 - INGEGNERIA INFORMATICA**

Anno regolamento: **2014**

CFU: **6**

Settore: **ING-INF/05**

Tipo Attività: **B - Caratterizzante**

Anno corso: **2**

Periodo: **II° semestre**

Sede: **PARMA**



Testi in italiano

Lingua insegnamento

Italiano

Contenuti

1. Elaborazione automatica dell'informazione
2. Calcolo parallelo
3. Calcolo distribuito
4. Sistemi peer-to-peer
5. Sistemi autonomici
6. Tecniche di simulazione
7. Computer quantistici

Testi di riferimento

M. Amoretti, dispense in inglese sugli argomenti del corso.
C. Ghezzi, D. Mandrioli, "Informatica Teorica", Città Studi, 1989.
D. E. Culler, J. Pal Singh, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufman, 1999.
B.P. Zeighler, H. Praehofer, T.G. Kim, "Theory of Modeling and Simulation", Second Edition, Academic Press, 2000.
M. Nielsen, I. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 2011.

Obiettivi formativi

La finalità del corso è quella di definire e caratterizzare i sistemi di elaborazione dell'informazione. In particolare verrà fornita allo studente la capacità di comprendere:

- modelli di programmazione parallela e distribuita
- sistemi di tipo client/server e sistemi di tipo peer-to-peer
- sistemi autonomici
- tecniche di simulazione
- computer quantistici

Lo studente imparerà ad applicare tali competenze per:

- sviluppare modelli di sistemi di elaborazione, in particolare per studiarne le prestazioni
- programmare sistemi paralleli e distribuiti
- progettare sistemi distribuiti di tipo client/server e di tipo peer-to-peer
- progettare sistemi di elaborazione dotati di caratteristiche autonome
- sviluppare simulazioni a eventi discreti
- progettare circuiti quantistici

Metodi didattici	<p>Il corso viene svolto con lezioni frontali in cui il docente spiega gli argomenti mostrando delle diapositive oppure scrivendo alla lavagna.</p> <p>La parte del corso dedicata ai sistemi peer-to-peer si conclude con un paio di esercitazioni pratiche in cui gli studenti sono invitati a utilizzare il middleware Sip2Peer per sviluppare semplici applicazioni. Le lezioni sulle tecniche di simulazione si concludono con un paio di esercitazioni pratiche in cui gli studenti sono invitati a utilizzare il tool di simulazione DEUS.</p>
-------------------------	---

Altre informazioni	<p>Il materiale didattico e di supporto alle lezioni è disponibile sul sito http://elly.dii.unipr.it</p>
---------------------------	---

Modalità di verifica dell'apprendimento	<p>Non sono previste prove in itinere.</p> <p>E' prevista una prova scritta (3 ore) articolata su 6-7 quesiti a risposta libera, relativi alla parte teorica del corso.</p> <p>E' previsto lo svolgimento di una tesina legata agli argomenti del corso. Tale tesina comporta, in genere, l'utilizzo o lo sviluppo di software, la stesura di una relazione e una presentazione alla commissione (con il supporto di diapositive). Ciascuna tesina può essere svolta da un singolo studente o da un gruppo di due studenti.</p>
--	---

Programma esteso	<p>Contenuti Sistemi di Elaborazione (42 ore)</p> <ol style="list-style-type: none"> 1. Elaborazione automatica dell'informazione <ol style="list-style-type: none"> 1.1. Cenni di teoria dell'informazione; 1.2. Tassonomia dei sistemi di elaborazione; 1.3. Macchine astratte; 1.4. Complessità computazionale 2. Calcolo parallelo <ol style="list-style-type: none"> 2.1. Architetture parallele; 2.2. Valutazione delle prestazioni; 2.3. Modelli di programmazione parallela; 2.4. Message Passing Interface (MPI); 2.5. MapReduce; 2.6. Sistemi Multicore, General Purpose GPU Programming, CUDA, OpenCL 3. Calcolo distribuito <ol style="list-style-type: none"> 3.1. Cluster computing; 3.2. Grid computing; 3.3. Cloud computing; 3.4. Pervasive computing 4. Sistemi peer-to-peer <ol style="list-style-type: none"> 4.1. Variabili di stato; 4.2. Dinamiche delle reti peer-to-peer; 4.3. Problematiche di progettazione; 4.4. Strategie di progettazione degli schemi di overlay; 4.5. Schemi di overlay (Napster, BitTorrent, eMule, Skype, Chord, Kademlia), Bitcoin 5. Sistemi autonomici <ol style="list-style-type: none"> 5.1. I quattro principi dell'autonomic computing; 5.2. MAPE-K; 5.3. NAM, NAM4J, Distributed Remodeling Framework 6. Tecniche di simulazione <ol style="list-style-type: none"> 6.1. Concetti generali sulle simulazioni; 6.2. Simulazioni ad eventi discreti 7. Computer quantistici <ol style="list-style-type: none"> 7.1. Cenni storici; 7.2. Richiami di algebra lineare; 7.3. Postulati della meccanica quantistica; 7.4. Quantum bit; 7.5. Circuiti quantistici; 7.6. Algoritmi quantistici
-------------------------	---



Testi in inglese

Lingua insegnamento	Italian
----------------------------	---------

Contenuti	<ol style="list-style-type: none"> 1. Computing systems 2. Parallel computing 3. Distributed computing 4. Peer-to-peer systems
------------------	--

- 5. Autonomic systems
- 6. Simulation techniques
- 7. Quantum computing

Testi di riferimento

M. Amoretti, lecture notes in english.
C. Ghezzi, D. Mandrioli, "Informatica Teorica", Città Studi, 1989.
D. E. Culler, J. Pal Singh, "Parallel Computer Architecture: A Hardware/Software Approach", Morgan Kaufman, 1999.
B.P. Zeighler, H. Praehofer, T.G. Kim, "Theory of Modeling and Simulation", Second Edition, Academic Press, 2000.
M. Nielsen, I. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 2011.

Obiettivi formativi

The main goal of the course is to define and characterize computing systems. In particular, the student will be provided with the ability to understand:

- parallel and distributed programming models
- client/server and peer-to-peer systems
- autonomic systems
- simulation techniques
- quantum computing

The student will learn to apply such a knowledge to:

- develop computing systems models, in particular for performance analysis
- program parallel and distributed systems
- design client/server and peer-to-peer distributed systems
- design autonomic systems
- develop discrete event simulations
- design quantum circuits

Metodi didattici

Lectures are given by the teacher, which illustrates the topics with the support of slides, or by writing on the blackboard. Lectures on peer-to-peer systems are completed by a tutorial on developing applications with the Sip2Peer middleware. Lectures about simulation techniques are completed by a tutorial on the DEUS simulation environment.

Altre informazioni

Lecture notes, slides and exercises are available on <http://elly.dii.unipr.it>

Modalità di verifica dell'apprendimento

There will not be exams during the course.

There will be a written exam (3 hours), with 6-7 open questions related to theoretical part of the course.

The students will have to work (alone or in couples) on a small project, usually requiring software use/development, the writing of a report and a final presentation.

Programma esteso

Lectures on Parallel and Distributed Computing (42 hours)

1. Computing systems

1.1. Information theory; 1.2. Taxonomy of computing systems; 1.3. Abstract machines; 1.4. Computational complexity

2. Parallel computing

2.1. Parallel architectures; 2.2. Performance evaluation; 2.3. Parallel programming models; 2.4. Message Passing Interface (MPI); 2.5. MapReduce; 2.6. Multicore systems, General Purpose GPU Programming, CUDA, OpenCL

3. Distributed computing

3.1. Cluster computing; 3.2. Grid computing; 3.3. Cloud computing; 3.4. Pervasive computing

4. Peer-to-peer systems

4.1. State variables; 4.2. Dynamics of peer-to-peer networks; 4.3. Design issues; 4.4. Design strategies for overlay schemes; 4.5. Overlay schemes

(Napster, BitTorrent, eMule, Skype, Chord, Kademia), Bitcoin

5. Autonomic computing

5.1. The four principles of autonomic computing; 5.2. MAPE-K; 5.3. NAM, NAM4J, Distributed Remodeling Framework

6. Simulation techniques

6.1. General concepts about simulation; 6.2. Discrete event simulation

7. Quantum computing

7.1. History of quantum computing; 7.2. Linear algebra; 7.3. The postulates of Quantum Mechanics; 7.4. Quantum bit; 7.5. Quantum circuits; 7.6. Quantum algorithms